

What is claimed is:

1 1. A method comprising:
2 dynamically obtaining one or more program operators from source code; and
3 applying data transformation to a portion of the source code based on one of said
4 one or more program operators to provide encrypting compiler-generated code.

1 2. The method of claim 1, including mixing the encrypting compiler-generated code
2 with the source code other than said portion before compilation.

1 3. The method of claim 1, further comprising deriving from the source code at least
2 one compiler-generated operator for said data transformation.

1 4. The method of claim 3, further comprising performing encryption using at least
2 one of said at least one compiler-generated operator and said at least one of said one or more
3 program operators.

1 5. The method of claim 1, further comprising selectively encrypting one or more
2 regions of the source code with a custom cipher formed from said at least one of said one or
3 more program operators.

1 6. The method of claim 1, including:
2 determining at least two references for each variable of a variable pair to
3 selectively encrypt and decrypt data in between said at least two references; and
4 associating at least two data values with each variable of the variable pair for
5 encryption of the data in a first transformation and decryption of the data in a second
6 transformation.

1 7. The method of claim 6, further comprising:
2 iteratively forming matching pairs of said data values for each variable of the
3 variable pair; and
4 creating interlocking Feistal networks in each iteration involving a different
5 matching pair of said data values.

1 8. The method of claim 7, further comprising:
2 enabling detection of usage of one or more redundant computations in the
3 interlocking Feistal networks; and
4 in response to a change in at least one of the one or more redundant computations,
5 provisioning for corruption of unrelated data values relative to said data values.

1 9. A method comprising:
2 analyzing flow of data in source code having one or more program operators to
3 determine matching references to a pair of variables;
4 determining a block of the source code in which said pair of variables is not used;
5 associating the matching references based on a heuristic to provide data
6 encryption to modify a portion of the source code into encrypting compiler-generated code; and
7 mixing the encrypting compiler-generated code with the source code.

1 10. The method of claim 9, wherein analyzing flow of data further including:
2 detecting a first region of the source code in which use of a stored value for at
3 least one variable of said pair of variables occurs; and
4 detecting a second region of the source code in which the stored value is defined
5 for the at least one variable of said pair of variables.

1 11. The method of claim 9, including utilizing the heuristic to enhance obfuscation of
2 the encrypting compiler-generated code within the source code using at least one of said one or
3 more program operators.

1 12. A method comprising:
2 identifying a first reference point and a second reference point within a set of
3 blocks of source code having one or more program operators;
4 associating an encryption code in proximity to the first reference point and
5 associating a decryption code in proximity to the second reference point; and
6 compiling a portion of the source code into encrypting compiler-generated code to
7 mix with the source code other than said portion.

1 13. The method of claim 12, further comprising:
2 customizing a cipher based on at least one of said one or more program operators;
3 selecting a block from the set of blocks, the block containing a first variable
4 having a maximum distance over the set of blocks, and a second variable having a next maximal
5 distance in the same block;
6 providing the encryption code to encrypt data in between a pair of references to
7 the first and second variables; and
8 providing the decryption code to decrypt said data.

1 14. The method of claim 13, further comprising recompiling the encrypting compiler-
2 generated code with the source code other than said portion into tamper resistant object code.

1 15. The method of claim 13, including:
2 deriving from the source code at least one compiler-generated operator for data
3 flow transformation; and

4 using at least one of said at least one compiler-generated operator and said at least
5 one of said one or more program operators to provide the encryption code.

1 16. An article comprising a medium storing instructions that, if executed enable a
2 system to:

3 dynamically obtain one or more program operators from source code; and
4 apply data transformation to a portion of the source code based on one of said one
5 or more program operators to form encrypting compiler-generated code.

1 17. The article of claim 16, further comprising instructions that if executed enable the
2 system to mix the encrypting compiler-generated code with the source code other than said
3 portion.

1 18. The article of claim 16, further comprising instructions that, if executed enable the
2 system to use at least one compiler-generated operator and said at least one of said one or more
3 program operators for encryption.

1 19. The article of claim 16, further comprising instructions that, if executed enable the
2 system to selectively encrypt one or more regions of the source code with a cipher formed from
3 said at least one of said one or more program operators

1 20. An apparatus comprising:
2 an analyzer to perform data flow analysis of source code to dynamically obtain
3 one or more program operators therefrom; and
4 a code transformer coupled to said analyzer to apply data transformation to select
5 a selected region of the source code in which to provide encrypting compiler-generated code
6 based on one of said one or more program operators.

1 21. The apparatus of claim 20, further comprising a cipher based on said at least one
2 of one or more program operators.

1 22. The apparatus of claim 20, further comprising an encryption engine to selectively
2 encrypt and decrypt the selected region based on references to a variable identified in the
3 selected region.

1 23. The apparatus of claim 22, further comprising a heuristic to select the selected
2 region and the references.

1 24. A system comprising:
2 a dynamic random access memory having source code stored therein;
3 an analyzer to perform data flow analysis of the source code to dynamically
4 obtain one or more program operators therefrom; and
5 a code transformer coupled to said analyzer to apply data transformation to select
6 a selected region of the source code to provide encrypting compiler-generated code based on one
7 of said one or more program operators.

1 25. The system of claim 24, further comprising a cipher based on said at least one of
2 one or more program operators.

1 26. The system of claim 24, further comprising an encryption engine to selectively
2 encrypt and decrypt the selected region based on references to a variable identified in the
3 selected region.

1 27. The system of claim 26, further comprising a heuristic to select the selected
2 region and the references.